

AFRL-SN-WP-TP-2003-107

**OZONE: A ZOOMABLE INTERFACE
FOR NAVIGATING ONTOLOGY
INFORMATION**



**Bongwon Suh
Benjamin B. Bederson**

FEBRUARY 2003

Approved for public release; distribution is unlimited.

Ó 2002 ACM

This work is copyrighted. The United States has for itself and others acting on its behalf an unlimited, paid-up, nonexclusive, irrevocable worldwide license. Any other form of use is subject to copyright restrictions.

**SENSORS DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7318**

REPORT DOCUMENTATION PAGE

OZONE: A Zoomable Interface for Navigating Ontology Information

ABSTRACT

We present OZONE (Zoomable Ontology Navigator), for searching and browsing ontological information. OZONE visualizes query conditions and provides interactive, guided browsing for DAML (DARPA Agent Markup Language) ontologies on the Web. To visually represent objects in DAML, we define a visual model for its classes, properties and relationships between them. Properties can be expanded into classes for query refinement. The visual query can be formulated incrementally as users explore class and property structures interactively. Zoomable interface techniques are employed for effective navigation and usability.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval – *query formulation, relevance feedback*; H.4.3 [Information Systems Applications]: Communication Applications – *information browser*

General Terms

Human Factors, Design, Language.

Keywords

Ontology, DAML, Browsing, Zoomable User Interface (ZUI), Jazz, WWW.

1. INTRODUCTION

Information on the World Wide Web (WWW) has expanded enormously during the last several years. Searching the Web by string matching and link analysis has been one of the most popular ways of finding information in the vast quantity of data. But since these approaches rely on syntactic information, the search result of such schemes is often limited and ineffective. The

Web has been designed for direct human processing. It is humans that write most of the web pages. Therefore, for accurate knowledge extraction, it is crucial to identify embedded semantic knowledge in them.

RDF (Resource Description Framework) and RDF schema are W3C recommendations to add metadata in order to turn the WWW into a machine-readable knowledgebase [3][11][16]. RDF offers a distinguished vocabulary to model classes, properties, and other basic schema primitives that can be referred to from its model. This model also can be extended to address sophisticated ontology representation techniques. An ontology is defined as shared formal conceptualization of a particular domain [6]. Practically, it can be regarded as a vocabulary and its definitions as well as relationships between concepts in it. Ontologies specify what concepts to represent and how they are related. They can be used to convey semantic information through machine-based communication and their description can be reused between similar domains. Using semantic information, the WWW will enable intelligent services such as information brokers, search agents and information filters which offer greater functionality and interoperability than current stand-alone services [6].

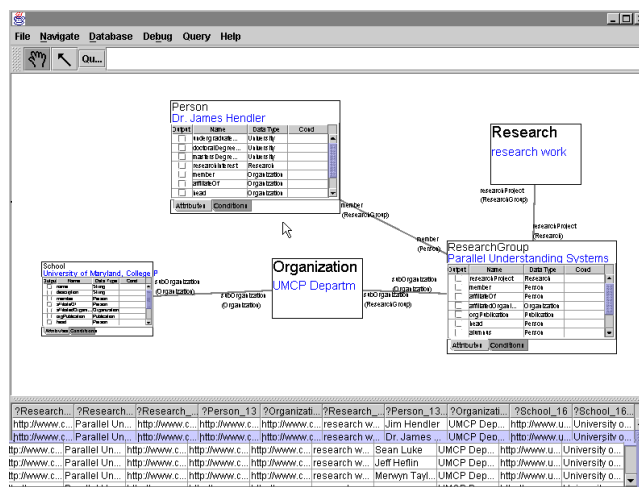


Figure 1. OZONE overview

DAML (DARPA Agent Markup Language) is one of the newly emerging standards for the metadata framework [5]. DAML is

based on RDF and extends it to facilitate agent-based computing. DAML allows communities to extend simple ontologies for their own use and also provides mechanisms for the explicit representation of semantic knowledge to augment web pages. Based on the RDF framework, every DAML individual is represented as a simple triple of subject, predicate and object. It might be easy for machines to interpret but, for humans, a more abstract model is needed.

2. EXAMPLE AND MOTIVATION

A typical query involves relations between multiple classes where some details are known and some aren't. As an example, assume that a user wants to find people working on a specific project. Further the user knows that the people who he/she is looking for work in a particular research group at the University of Maryland. But the user does not know about the organization of research groups.

In the case that a user does not know much about the ontology structure, it is very difficult to form a valid query. The user has to know the names and semantic meanings of classes, properties and their relationships precisely. Furthermore, ontology structures are not well-formed, which confuses the user and they are distributed on the WWW. These aspects hinder efficient query formulation.

To clearly demonstrate the issues surrounding this example, we define a specific ontology that we will use throughout this paper.

Table 1. Example ontology

Class	Property	Type of Property
Person	member	Organization
ResearchGroup	researchProject	Research
	member	Person
	subOrganization	Organization
Organization	subOrganization (have a sub-organization of)	Organization
	subOrganization (is a sub-organization of)	Organization
	member	Person
Research	researchProject	ResearchGroup
School	subOrganization	Organization

Now we can formally state the query first introduced using the ontology in Table 1. The text-based RDF query languages such as Squish [14], RDFDB QL [8] are proposed to form complex queries efficiently. The following is an example pseudo-Squish statement to get an answer for the previous question.

Practically, classes may be thought of as a *vocabulary* that represents semantic meanings for a set of objects, or object groups that share common attributes. For example, the “Person” is a class in our running example. Properties may be regarded as *attributes* of classes and in this sense correspond to traditional attribute-value pairs. For instance, the “member” is a property of the “Person” class. But, in the DAML model, the value of a property can be typed and linked with other classes. As in Table 1, the “member” property of the “Person” class is defined as “Organization” type and it should have values of only that type.

4.2 Visual Model

In OZONE, a class node is defined as an aggregation of its name and properties that have relationship with the class and represented as shown in Figure 2.

As stated above, each property can have class types and, therefore, classes can be induced from those typed properties. For example, the “Organization” class can be induced from the “member” property of the “Person” class. This procedure is called “*property expansion*” because a class is expanded from a property and OZONE uses this as a major query formulation method. Figure 2 shows the expansion of the “ResearchGroup” from the “member” property of the “Person” class. In Figure 3, the “ResearchGroup” has multiple property expansions.

For properties which don’t have specific types, OZONE supports general string matching query formulation as shown in Figure 3. The “name” property of “School” in Figure 3 has no type definition and is regarded as “String” type as default. Accordingly only string input is accepted as a valid query operation.

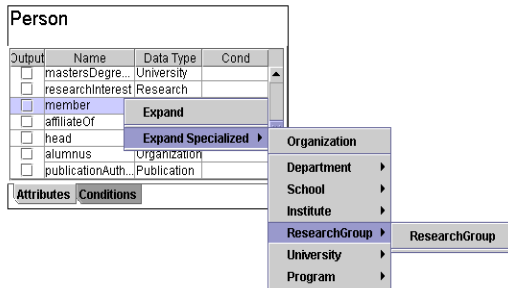


Figure 2. Class node and property expansion

By providing a list of properties for each class, OZONE helps users to understand the meaning of classes and also provides intuitive and effective feedbacks. The class visual node is displayed in zoomable space so that it can be moved, zoomed in, zoomed out freely.

Properties of a class are collected from DAML definitions that are scattered over the Web. OZONE parses and extracts DAML ontologies and retains the information inside until the end of its session. To be scalable, internal ontology representations require more efficient management, which is not implemented yet.

4.3 Links and Browsing

Relationships in DAML are modeled as links in OZONE and are used to facilitate query refinement. A link is created as a result of the ‘property expansion’ as mentioned before.

Expanding a property that originated from property p in class c implies adding query conditions to class c as follows:

$$E(c, p) = \{ \langle o, o' \rangle \mid o \text{ is an instance of } c, \\ \exists o' \text{ is an instance of } c' \text{ such that } \langle o, o' \rangle \in p \}$$

Suppose that there is a visual class node drawn on the screen without any conditions. Since there is no condition, every instance matching the class will be returned as a result, yielding a large result set. For example, the “Person” node without any link would return all persons in a knowledgebase as its result.

Property expansion can be used to narrow the query result by adding conditions to a query. For example, expanding the ‘member’ property filters out instances of the “Person” that do not have any affiliation. Instances of ‘Person’ that have relationships with ‘Organization’ instances are included in the result set. The property expansion can be applied multiple times as a query becomes complicated.

In addition to the property expansion, specifying a string value for a literal property is another way of adding a query condition. As an instance, editing the “name” property of the “School” class as “Maryland” chooses “School” instances whose name contains the literal. Currently, string matching is the only way to check equality.

Multiple query conditions are connected conjunctively to produce the result. Because of this strategy, OZONE is not able to specify disjunctive conditions.

Browsing begins with searching a class from the class list or the class hierarchy tree which shows all searchable classes in a hierarchical format. Once a class is selected, it is put on the screen as a visual node. Query conditions can be added to that node either by inputting a string or by property expansion. During this procedure, a user can execute the query whenever he/she wants to see the intermediate results. After the intermediate query is executed, its results are shown at the bottom of the screen in a table. Retrieved class information in the table is remapped into class nodes when a row in the table is selected as blue labels under the class title. Double clicking each entry in the result table will launch a web browser with its value if it is a URL. Figure 3 shows an example graph that is drawn by property expansions.

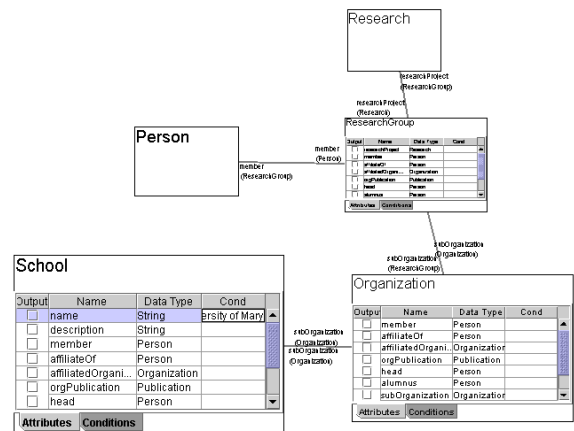


Figure 3. This represents the query refinement by property expansion and specifying string value.

4.4 Specialization and Generalization

DAML ontologies are hierarchical. So we can take advantage of the hierarchy when querying. If a search result is too big, OZONE

can narrow the result by specialization. For example, searching for ‘Organization’ can be specialized into ‘School’. It will exclude instances such as ‘Company’, ‘Government’, and so on. Generalization is the opposite concept to specialization. When general results are needed, the query condition is loosened up by generalization. For example, ‘Student’ can be generalized into ‘Person’. The query result would include instances such as ‘Faculty’, ‘Staff’, and so on as well as ‘Student’.

These operations are accessed with two mechanisms in OZONE. First, they can be applied by property expansion. The subclass of an expandable property type can be expanded instead of the original property type. As in Figure 2, the “ResearchGroup” class is expanded instead of its superclass “Organization” from the “Person” class so that persons in a research group can be retrieved. Secondly, generalization and specialization can be applied to a visual node itself. Clicking the header of a class node will show popup menu that contains the list of super classes and subclasses of the class. By adjusting the level of coverage, users can narrow or widen the result.

The hierarchy structure in ontologies is pre-defined by the ontology designers who are experts in their domain and ontologies. Since the generalization and specialization are guided by ontologies, more accurate querying is possible than with non-guided searching.

4.5 Query Abstraction by Grouping

In OZONE, any sub-graph can be grouped and transformed into a single node by choosing the ‘Group’ menu in the main menu after selecting nodes on the screen. The collection of nodes is zoomed out and a simple new node replaces the collection. Users can access the detailed sub nodes at any time by zooming in.

This feature offers a general way of abstracting queries. High-level queries can be formulated in simpler forms by abstracting low-level details. A group node is also useful when importing queries. Even if users do not know the structure inside a group, they can use the group node the same way as they use a regular class node. Users can add more conditions on this group node either by property expansion or by specifying string values.

The grouping also helps to reduce the number of objects on screen. Figure 4 shows an example animation sequence in which five individual visual nodes are abstracted into a single node.

Figure 4. Grouping multiple nodes into a single node

4.6 User Interface

OZONE provides basic editing functions such as deleting, moving, centering, zooming in, zooming out, grouping, saving, and loading. Individual nodes and group nodes as well as the entire view can be zoomed in and out independently without restriction according to users’ preferences. This makes it possible to avoid the screen being crowded by too many objects. All zooms and transitions are animated so users can maintain a sense of object constancy.

At any scale, double clicking on a class node zooms it into its original size and moves it to the center of the screen. Inputting

searching(n)-.8267 rht3.5(le)80s too m8(rp9.6(h)dpe.5(c)9.6(h)t(3.5(too m8(re.5)

5. FUTURE WORK AND CONCLUSION

Since OZONE is in its early stage, there is much room for improvement. The major improvements will be to merge searching and browsing with authoring. Adding new ontology information is like solving a jigsaw puzzle because most of newly authored information is created as an extension from existing context information that is scattered over the Web. It is crucial to identify relationships between ontology artifacts when creating new instances. Another desirable improvement would be automatic suggestions when searching a class. Since ontologies are defined by ontology designers, keywords specified by casual users do not necessarily match the class names defined in ontology vocabularies. By comparing users' searching keywords with words in WordNet [18] and suggesting the closest classes in ontology, this kind of confusion might be avoided. We have only started our work in using OZONE with real users. We plan on performing usability studies with representative users doing realistic tasks and refining the interface based on their feedback.

OZONE is an interface for navigating ontology-based knowledgebases of which schematic information is gathered from the Web. It visualizes classes and properties of DAML to facilitate efficient query formulation. The DAML ontology framework provides semantic layers on web pages so that search agents can find information accurately and effectively. But searching information in an ontology data set is not easy because it does not have well formed structure and logical expressions should be used to represent relationships between individuals in ontologies.

OZONE illustrates an ontology query that is unlike the query interfaces that requires users to remember detailed ontology information. Classes are provided with property information so that users can specify relationships with other classes by clicking on a property in a visual class. Specialization and generalization is another way of query refinements and is also supported by menus. Class hierarchies in ontologies are provided to users during query formulation to narrow down or broaden query results. In OZONE, query conditions can be abstracted. Multiple class nodes are shrunk into a single node for efficient screen usage. It also provides high-level views for complex queries.

Our experience with designing and developing OZONE for navigating ontology information has highlighted a number of new issues. First, authoring ontology information is a main bottleneck for facilitating semantic services. Second, browsing can conflict with searching. It is hard to form complex queries by browsing while it is not easy to provide users with available options when searching. Allowing both functionalities without sacrificing usability needs to be researched further. Finally, an effective user interface dealing with ontology information can impact the development of intelligent services on the WWW.

6. ACKNOWLEDGEMENT

We would like to thank Dr. James Hendler and Jeff Heflin for their support, suggestions, ideas, and technical help. And thanks to Lance Good who have commented on implementation issues. This work has been supported in part by DARPA's Command Post of the Future project, contract #F336159711018.

7. REFERENCE

- [1] Bederson, B.B., Hollan, J.D., Perlin, K., Meyer, J., Bacon, D., Furnas, G. W., "Pad++: A Zoomable Graphical Sketchpad

for Exploring Alternate Interface Physics," *Journal of Visual Languages and Computing*, 7, 3 – 31, 1996

- [2] Bederson, B.B., Meyer, J., and Good, L., "Jazz: An Extensible Zoomable User Interface Graphics ToolKit in Java," In *Proc. of User Interface and Software Technology (UIST 2000)*, ACM Press, 2000
- [3] Brickley, D. and Guha, R., "Resource Description Framework (RDF) Schema Specification," W3C <http://www.w3c.org/TR/2000/CR-rdf-schema-20000327/>, 2000
- [4] Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam K., and Slaterry, S. "Learning to Extract Symbolic Knowledge from the World Wide Web," In *Proc. of the 15th National Conference on Artificial Intelligence (AAAI-98)*, 1998
- [5] DAML, <http://www.daml.org>
- [6] Decker, S., Melnik, S., Van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., Erdmann, M. and Horrocks, I., "The semantic web: The roles of XML and RDF," *IEEE Internet Computing* Sept.-Oct. 2000.
- [7] Fensel, D., Angele, J., Decker, S., Erdmann, M., Schnurr, H.P., Staab, S., Studer, R. and Witt, A., "On2broker: Semantic-based Access to Information Sources at the WWW," In *Proc. of the World Conference on the WWW and Internet (WebNet 99)*, Honolulu, Hawaii, USA, October, 1999
- [8] Guha, R. V. RDFDB QL, <http://web1.guha.com/rdfdb/query.html>
- [9] Heflin, J. and Hendler, J. "Searching the Web with SHOE," *AAAI-2000 Workshop on AI for Web Search*, 2000.
- [10] JAXP XML parser, Sun Microsystems <http://java.sun.com/xml>
- [11] Lassila, O. and Swick, R. "Resource Description Framework (RDF) Model and Syntax," W3C, <http://www.w3c.org/TR/1999/REC-rdf-syntax-19990222>, 1999.
- [12] Luke, S., Heflin, J., "SHOE 1.0," Proposed Specification, <http://www.cs.umd.edu/projects/plus/SHOE/spec.html>, 1998.
- [13] McGuinness, D. L., Fikes, R., Rice, J., and Wilder, S., "An Environment for Merging and Testing Large Ontologies," In *Proc. of the 7th International Conference on Principles of Knowledge Representation and Reasoning (KR2000)*, 2000.
- [14] Miller, L., "RDF querying using Squish," <http://swordfish.rdfweb.org/rdfquery/>
- [15] Shneiderman, B., *Designing the User Interface*, 3rd ed. Addison-Wesley Publishing Co., MA, 1998.
- [16] Staab, S., Erdmann, M., Mädche, A., and Decker, S., "An Extensible Approach for Modeling Ontologies in RDF(S)," In *Proc. of ECDL 2000 Workshop on the Semantic Web*, 11-22, 2000.
- [17] Stoffel, K., Taylor, M., and Hendler, J., "Efficient Management of Very Large Ontologies," In *Proc. of the 14th National Conference on Artificial Intelligence (AAAI-97)*, 1997.
- [18] WordNet, <http://www.cogsci.princeton.edu/~wn/>